Mixed-Dense Connection Networks for Image and Video Super-Resolution

Kuldeep Purohit*, Srimanta Mandal, A. N. Rajagopalan

IPCV Lab, Department of Electrical Engineering, Indian Institute of Technology Madras, TN, India

Abstract

Efficiency of gradient propagation in intermediate layers of convolutional neural networks is of key importance for super-resolution task. To this end, we propose a deep architecture for single image super-resolution (SISR), which is built using efficient convolutional units we refer to as mixed-dense connection blocks (MDCB). The design of MDCB combines the strengths of both residual and dense connection strategies, while overcoming their limitations. To enable super-resolution for multiple factors, we propose a scale-recurrent framework which reutilizes the filters learnt for lower scale factors recursively for higher factors. This leads to improved performance and promotes parametric efficiency for higher factors. We train two versions of our network to enhance complementary image qualities using different loss configurations. We further employ our network for video super-resolution task, where our network learns to aggregate information from multiple frames and maintain spatio-temporal consistency. The proposed networks lead to qualitative and quantitative improvements over state-of-the-art techniques on image and video super-resolution benchmarks.

Keywords: Super-resolution, Deep Learning, Residual Networks, Dense connection, video super-resolution.

1. Introduction

Single image super-resolution (SISR) aims to estimate a high-resolution (HR) image from a low-resolution (LR) input image, and is an ill-posed problem. Due to its diverse applicability starting from surveillance to medical diagnosis, and from remote sensing to HDTV, the SISR problem has gathered substantial attention from computer vision and image processing community. The ill-posed nature of the problem is generally addressed by learning a LR-HR mapping function in a constrained environment using example HR-LR patch pairs.

One way is to learn a mapping function that linearly correlates the HR-LR patch pairs. Such linear functions can be easily learned with few example images as has been practiced by some SR approaches [1, 2, 3]. But, linear mapping between such patch pairs may not be representative enough to learn different complex structures present in the image. The mapping function would benefit from learning non-linear relationships between HR-LR patch pairs. Recent convolutional neural network (CNN) based models are quite efficient for such a purpose, and can be useful

*Corresponding author. Tel.: +91-9884-264834

Email addresses: kuldeeppurohit3@gmail.com (Kuldeep Purohit), in.srimanta.mandal@ieee.org (Srimanta Mandal), raju@ee.iitm.ac.in (A. N. Rajagopalan)

in extracting relevant features by making deeper models. However, deeper models often face vanishing/exploding gradient issues, which can be partially mitigated by using residual mapping [4, 5]. Deep residual models has been employed for higher level vision tasks, where batch normalization is generally used for a useful class-specific normalized representation. However, such representation is not much useful in low-level vision task such as SR [6]. Most deep CNN based SR models do not make full use of the hierarchical features from the original LR images. Thus, the scope of improvement in performance is there in effective employment of the hierarchical features from all the convolutional layers, as has been employed by a residual dense network [7] using a sequence of residual dense blocks (RDBs). However, most of these deep networks require huge number of parameters, which becomes a bottleneck in the situation where limited computational resources are available.

Further, most of the networks need to be trained separately for different scale factors. This issue can be addressed by jointly training the model for different scale factors, as has been performed by VDSR [8]. Moreover, VDSR requires bicubic interpolated version of the LR image as an input. Processing interpolated images precludes the model from learning direct LR-HR feature mapping. Additionally, passing the high-resolution image and feature-maps through a large number of layers increases the memory consumption as well as computational requirements. Another way to consider multiple scale factors is to train a network jointly for different factors by including scale-specific features [6]. However, these techniques produce sub-optimal results for higher scale factors like 8.

Motivated by the performances of residual connections and dense connections, we propose a deep architecture for single image super-resolution (SISR) that consists of efficient convolutional units, which we entitle as mixed-dense connection blocks (MDCB). MDCB combines the advantages of residual and dense connections while subduing their limitations. The combination improves the flow of information through the network, alleviating the gradient vanishing problem. In addition, it allows the reuse of feature maps from preceding layers, avoiding the re-learning of redundant features. In order to master our network for super-resolving different scale factors, we make use of weight transfer strategy via scale-recurrent framework. Intuitively, filters learnt for smaller scale factors can be transfered to higher-ones. Sharing of parameters across scales is crucial for efficiently super-resolving by higher up-sampling factors. Our scale recurrence framework built using MDCBs is parametrically more efficient than most of the existing works, enabling our strategy to work with limited resources.

It has been recently found that a result with good RMSE value often fails to satisfy perceptually, and the converse also holds true [9]. Thus, to obtain photo-realistic results, we employ a GAN framework with deep feature loss (VGG) function in the network, which leads to a second network. These two networks enable us to traverse along the perception-distortion curve. The first network is trained to produce an output with better RMSE score, whereas the second one tries to produce result with better perceptual score. Different weighting schemes of GAN and VGG and pixel reconstruction losses enables us to traverse along the perception distortion curve [9] and

can be used to reach a desirable trade-off between the two.

We have further employed our network for the task of video SR, where our model super-resolves each frame by aggregating HR information from a local temporal window of LR frames. We demonstrate that the proposed SR framework can approximate the inverted image formation process, while maintaining spatio-temporal consistency and the estimated HR frames are good candidates for representing the ground truth frames of a video. The proposed networks help in achieving better qualitative and quantitative performance against the state-of-the-art techniques on image and video super-resolution benchmarks.

The rest of the paper is divided as follows: Section 2 discusses the related works and highlights our contributions. Section 3 describes the proposed architecture and the motivation behind it. Section 4 illustrates the concept of perceptual and objective quality trade-off during SR. Proposed network is extensively evaluated for SISR in Section 5. Section 6 contains analysis of various training configurations of the proposed network. Extension of our network to video SR is presented in Section 7 along with experimental results. Section 8 concludes the paper.

2. Related Works and Contributions

We address the problems of SR for single image and extend it to video. To discuss the related works for each of the problems, we discuss this section mainly in two parts, corresponding to the single image and video SR. In each category, there are numerous approaches starting from conventional to deep learning based. Though, our technique is based on deep learning, we also brief some of the conventional techniques for completeness.

2.1. Works on Single Image SR

Super resolving single image generally requires some example HR images to import relevant information for generating the HR image. Two stream of approaches make use of the HR example images in their frameworks: i) Conventional approaches, and ii) deep learning based approaches.

2.1.1. Conventional Single Image SR

Conventionally, single image SR approaches work by finding out patches similar to the target patch in the database of patches, extracted from example images. However, the possibility of many similarities along with the imaging blur, and noise often makes the problem ill-posed. Different prior information help in address the ill-posed nature of the problem.

Natural images are generally piecewise smooth, hence it is often incorporated as prior knowledge by means of Tikhonov, total variation, Markov random field, etc. [10, 11, 12]. Image patches tend to repeat in the image non-locally, and can provide useful information in terms of non-local prior to super-resolve an image [13, 2, 14, 3]. The redundancy present in an image indicates sparsity in some domain, and can be employed in SR framework by employing sparsity inducing norm. Here, the assumption is that a target patch can be represented by combining few

patches from the database linearly[1, 15, 2, 3]. Different priors can be combined to provide various information to the framework. For example, sparsity inducing norm can be combined with non-local similarity to improve the SR performance [16, 3].

Although, the sparsity based prior works quite efficiently, the linear mapping of information fails to represent complex structures of image. Here, deep-learning based approaches behave better through non-linear mapping of HR information [17, 18, 19, 8, 20, 21, 22, 23, 24, 6, 25, 26].

2.1.2. Deep Learning Based Single Image SR

Deep learning stepped into the field of SR via SRCNN [27], which extends the concept of sparse representation using CNN. CNNs typically contains large number of filters along with series of non-linearities and hence are a better candidate for representing complex input-output mapping than the conventional approaches, and are shown to yield superior results. However, increasing the depth of such architecture increase difficulty in training. Introducing residual connections into the framework along with skip connections and/or recursive convolutions are known to make it less cumbersome. Following such methodologies, VDSR [8] and DRCN [28] have demonstrated performance improvement. The power of recursive blocks involving residual units to create a deeper network is explored in [20]. Recursive unit in conjunction with gate unit acts as a memory unit that adaptively combines the previous states with the current state to produce a super resolved image [24]. However, these approaches interpolate the LR image to the HR grid before feeding it to the network. This technique increases the computational requirement since all the convolution operations are then performed on high-resolution feature-maps. To alleviate this computational burden, networks have been tailored to extract features from the LR image through a series of layers. Towards the end of such networks, up-sampling process is performed to match with the HR dimension [18, 5]. This process can be made faster by reducing the dimension of the features going to the layers that maps from LR to HR, and is known as FSRCNN [18].

Recent studies show that traditional metrics used to measure image restoration quality do not correlate with perceptual quality of the result. The work of in SRResNet [5] utilized ResNet [4] based network with adversarial training framework (GAN) [29] with perceptual loss [30] to produce photo-realistic HR results. The perceptual loss is further used with texture synthesis mechanism in GAN based framework to improve SR performance [23]. Though these approaches are able to add textures in the image, sometimes the results contain artifacts. The model architecture of SRResNet [5] is further simplified and optimized to achieve further improvements in EDSR [6]. This is further modified in MDSR [6], which performs joint training for different scale factors by introducing scale-specific feature extraction and pixel-shuffle layers, while keeping rest of the layers common.

2.2. Works on Video SR

We briefly discuss some of the related works on video SR including conventional approaches and recent deep learning based approaches.

2.2.1. Conventional Approaches

The seminal work of [31] has enabled various development in multi-frame SR. Motion between the consecutive frames can be employed in reconstruction based methods [32, 33, 34]. The quality of the super-resolved video depends on the accuracy of motion estimation of pixels. In order to maintain the continuity of the result, a back-projection method was introduced to minimize reconstruction error iteratively [35]. Example-based image SR approaches cannot be directly extended for video by super-resolving each frame separately. Independent processing of frames can give rise to flickering artifacts among the frames. This issue can be mitigated by introducing smoothness prior among the frames [36]. Different spatial-temporal resolutions of frames can also be combined to super-resolve a video [37], where a dictionary has been constructed by using scene specific HR images captured by a still camera. Single image SR has also been exercised for video SR in frame by frame basis by incorporating inverted image formation process [38].

The accurate motion estimation requirement for video SR has been relaxed by multidimensional kernel regression [39]. Here each pixel is approximated by 3D local series, whose coefficients are estimated by solving a least-square problem, where weights were introduced based on 3D space-time orientation in the pixel neighborhood [39]. However, this method still assumes the blur kernel that was involved in the LR image formation model. To do away with the assumptions of blur kernel along with motion and noise level a Bayesian strategy has been developed [40], where the degradation parameters are estimated from the given image sequence. Motion blur due to relative motion between camera and object can be handled by considering the least blurred pixels through an EM framework [41].

2.2.2. Deep Learning Based Approaches

The motion information among the frames has been explored by BRCN [42], which modeled long-term contextual information of frames using recurrent neural networks. Three types of convolutions (feed-forward, recurrent, and conditional) were involved to meet with requirement of spatial dependency, long-term temporal and contextual dependencies. However, iterative motion estimation procedure increases the computational burden. To reduce the computing load, DECN [43] introduced non-iterative framework, where different hand-crafted optical flow algorithms are used to generate different SR versions, which were finally combined using a deep network. Similar technique was employed in VSRnet [44], where motion was compensated in the LR frames by using an optical flow algorithm. The pre-processed LR frames were sent to a pre-trained deep network to generate the final result. Real-time estimation of motions among the input LR frames was tried out by VESPCN [45], which is an end-to-

end deep network that extracts the optical flow by warping frames using a spatial transformer [46]. Here, the HR frames are produced by another deep network.

Motion estimation and compensation has been practiced in many video SR approaches. However, Liu et al. [47] have demonstrated that adaptive usage of the motion information in various temporal radius of a temporal adaptive neural network can produce better results. Several inference branches were used for each temporal radius. The resultant images from these branches are assembled to produce the final result [47]. The motion compensation module of VESPCN [45] was employed in [48], where sub-pixel motion compensation layer was introduced to perform simultaneous motion compensation and resolution enhancement. After the motion compensation, the frames are super-resolved using an encoder-decoder framework with skip connections [49] and ConvLSTM [50] module for faster convergence as well as to take care of the sequential nature of video. Previous end-to-end CNN based video SR methods have focused on explicit motion estimation and compensation to better re-construct HR frames. Unlike the previous works, our approach does not require explicit motion estimation and compensation steps.

2.3. Contributions

The contributions of our work are:

- We present Mixed Dense Connection Network which is composed of building blocks designed to utilize the strengths of both residual and dense connections, while addressing their limitations. Our network performs favorably against state-of-the-art methods for different scale factors while being parametrically efficient.
- We exploit unconventional training configurations, namely scale-recurrent configuration for efficiently addressing multiple scale-factors, and variable-depth configuration to perform resource-aware super-resolution.
- We analyze the effect of adversarial (GAN) and CNN feature-loss (VGG) for training our network, to achieve trade-off between MSE and perceptual score. We show that different weights to GAN and VGG losses lead to different points in the perception-distortion plane.
- We further extend the proposed network to facilitate video super-resolution, yielding state of-the-art performance on standard benchmark.

3. Architecture Design

Success of recent approaches has emphasized the importance of network design. Specifically, the most recent image and video SR approaches are built upon two popular image classification networks: residual networks [4] and densely connected networks [51]. These two network designs have also enjoyed success and achieved state-of-the-art performance in other image restoration tasks such as image denoising, dehazing and deblurring. Motivated

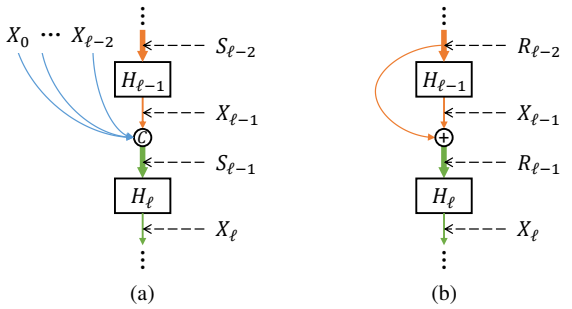


Figure 1: (a) A Dense connection, (b) A Residual connection. $H(\cdot)$ is a non-linear transformation function, X is feature map, S is the result of connection function C, and R represents the feature map after skip connections [54].

by the generalization capability of such advances in network designs, our work explores further improvements in network engineering which enable efficient extraction of high-resolution features from low-resolution images.

3.1. Dense Connection Topology

The work of [52] bridged the densely connected network [51] and residual networks with higher order recurrent neural networks [53] to provide a new understanding of the densely connected network. Here, we summarily describe the mathematical descriptions of Dense-block and Residual-block, which demonstrates that they belong to the same fundamental "dense topology" [52, 54] which can be defined as a path topology in which each layer is connected with all the previous layers.

Consider a network that comprises of L layers; each of which implements a non-linear transformation $H_\ell(\cdot)$, where ℓ indexes the layer and $H_\ell(\cdot)$ is a composite function of several operations which could include linear transformation, convolution, activation function, pooling, batch normalization etc. As illustrated in Fig. 1 (a), X_ℓ refers to the immediate output of the transformation $H_\ell(\cdot)$ and S_ℓ is the result of the connection function $C(\cdot)$ whose inputs come from all the previous feature-maps X (i.e., $X_0, X_1, ..., X_\ell$). Initially, S_0 equals X_0. General form of "dense topology" is defined as a path topology in which each layer H_ℓ is connected with all the previous layers $H_0, H_1, ..., H_{\ell-1}$ using the connection function $C(.)$ as:

$$X_\ell = H_\ell(C(X_0, X_1, \cdots, X_{\ell-1})). \tag{1}$$

For DenseNet [51], the input of ℓ^{th} layer is the concatenation of the outputs $X_0, X_1, ..., X_{\ell-1}$ from all the preceding layers. Therefore, we can write DenseNet as

$$X_\ell = H_\ell(X_0 \parallel X_1 \parallel \cdots \parallel X_{\ell-1}) \tag{2}$$

where $\parallel$ refers to the concatenation operation. As shown in Eqn. 1 and Eqn. 2, DenseNet directly follows the formulation of "dense topology", whose connection function is the pure concatenation (Fig. 1(a)).

Given the standard definition from [55], ResNet poses a skip connection that bypasses the non-linear transfor-

mations $H_\ell(\cdot)$ with an identity mapping as:

$$R_\ell = H_\ell(R_{\ell-1}) + R_{\ell-1}, \tag{3}$$

where R refers to the feature-maps directly after the skip connection (Fig. 1(b)). Initially, R_0 equals X_0. Now we concentrate on R_ℓ which is the output of $H_\ell(\cdot)$ as well

$$X_\ell = H_\ell(R_{\ell-1}). \tag{4}$$

By substituting Eqn. 3 into Eqn. 4 recursively, we can rewrite Eqn. 4 as:

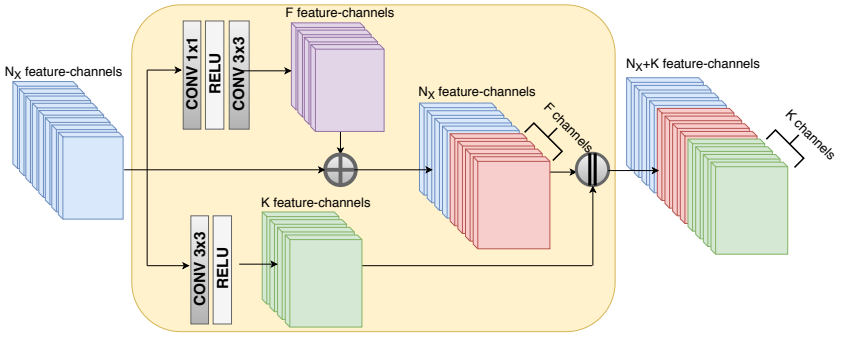
$$\begin{aligned} X_\ell &= H_\ell(R_{\ell-1}) = H_\ell(H_{\ell-1}(R_{\ell-2}) + R_{\ell-2}) = H_\ell(H_{\ell-1}(R_{\ell-2}) + H_{\ell-2}(R_{\ell-3}) + R_{\ell-3}) = \cdots \\ &= H_\ell(\sum_{i=1}^{\ell-1} H_i(R_{i-1}) + R_0) = H_\ell(\sum_{i=1}^{\ell-1} X_i + X_0) = H_\ell(X_0 + X_1 + \cdots + X_{\ell-1}). \end{aligned} \tag{5}$$

As shown in Eqn. 5, the output $R_{\ell-1}$ of ResNet can be represented as the element-wise sum of all the previous layers – $X_0, X_1, ..., X_{\ell-1}$. This implies that ResNet is actually identical to a form of "dense topology", where the connection function $C(\cdot)$ is the element-wise addition of feature-maps (Fig. 1(b)).

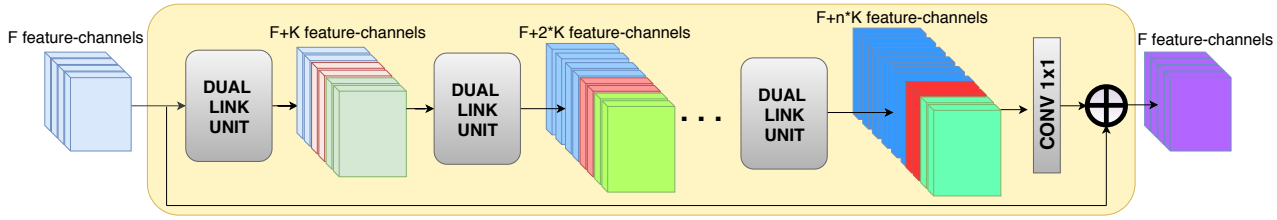
3.2. Proposed Architecture

The effectiveness of residual and dense connections has been proved via the significant success in various vision tasks yet, they cannot be considered as the optimum topology. For example, too many additions on the same feature space may impede the information flow in ResNet [51], and there may be the same type of raw features from different layers, which leads to a certain redundancy in DenseNet. Recent works of [52, 54] partially addressed these issues and demonstrate improvement in image classification performance. However, very few works have explored optimal network topologies for low-level vision tasks (e.g., image and video SR). We build upon the understanding of dense-topology to design a network for the task of super-resolution that benefits from a mixture of such connections and term it as Mixed-Dense Connection Network (MDCN).

Our MDCN not only achieves higher accuracy but also enjoys higher parameter efficiency than the state-of-the-art SR approaches. Its strength lies in its building blocks called Mixed-Dense Connection Blocks (MDCBs) which contain a rich set of connections to enable efficient feature-extraction and ease the gradient propagation. Inclusion of addition and concatenation based connections improves classification accuracy, and is more effective than going deeper or wider, given an option to increase the capacity of the network. In each MDCB, n Dual Link Units are present. Additive links in the unit grant the benefits of reusing common features with low redundancy, while concatenation links give the network more flexibility in learning new features. Although additive link is flexible with their positions and sizes, each Dual Link Unit performs the additive operation to the last F features of



(a) Structure of our *Dual Link* unit. The first link selectively modifies F existing feature-maps through addition, while the second link adds K new features through concatenation.



(b) Design of our mixed-dense connection block (MDCB). The features channels in red-shade correspond to the channels modified by the additive connection. The features channels in green-shade correspond to the channels added due to concatenation connection.

Figure 2: Structural details of our mixed-dense connection block (MDCB). Each block accepts the darker shade in the feature channels corresponds to deeper and rich features extracted from the initial feature map.

the input. Within each unit, the number of features for additive connections is F and the concatenating connections is K. A visual depiction of these connections can be seen in the Fig. 2(a). This unit adds K new feature maps to the input. The number of features in the input to each MDCB is F as shown in Fig. 2(b) and after n units, the feature maps contains $F + n * K$ channels. The growth-rate K of concatenation connections was shown to affect the image classification performance positively in [52, 54]. Further, it was experimentally demonstrated in [7] that deep networks containing many dense blocks stacked together are difficult to train for image restoration and result in poor performance. To handle this, we utilize a gating mechanism to allow larger growth rate by reducing the number of features and hence stabilizes the training of wide network. Each convolution or deconvolution layer is followed by a rectified linear unit (ReLU) for nonlinear mapping, except for the final 1×1 layer.

In summary, the advantage of mixed-dense connections manifests in form of significant improvement in propagating error gradients during training. The block has two advantages: Firstly, existing feature channels get modified to learn the residuals: helping in deeper and hierarchical feature extraction. Secondly, feature concatenation with moderate growth-rate promotes new feature exploration. While having these advantages, it also handles the disadvantages: moderate growth rate leads to reduction of redundant features.

Our complete network (MDCN) broadly consists three parts: initial feature extraction module, a series of mixed-dense connection block (MDCBs) and an HR reconstruction module, as shown in Fig.3. Specifically, the feature extraction module contains two convolutional layers to extract basic feature-maps containing F channels,

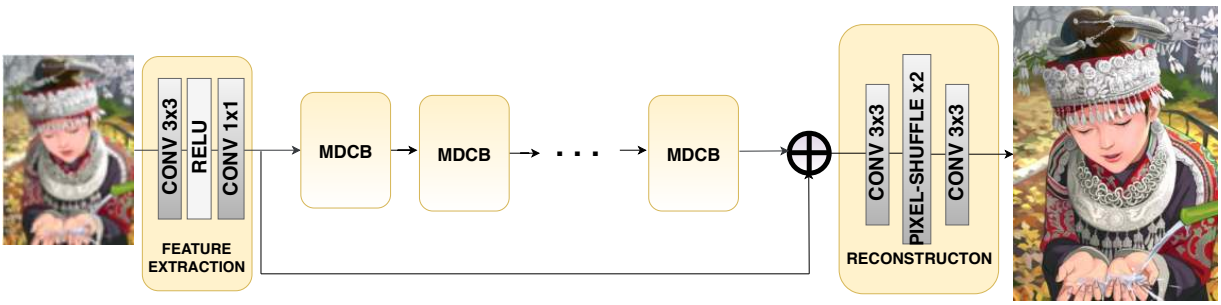


Figure 3: Architecture of our Mixed Dense Connection Network (MDCN).

which are fed to the first MDCB. The HR reconstruction module contains a convolution layer which takes F chanenls features in LR space to $4 * F$ channels. The pixel-shuffle layer accepts $4 * F$ channels in LR space and returns F channels in HR space, which are passed to the final conv layer to obtain the 3-channel HR image.

Initial upsampling: A few approaches interpolate the original LR image to the desired size to form the input to the network. This pre-processing step not only increases computation complexity, but also loses some details of the original LR image. Our network operates on the LR image, extracting deep hierarchical features from it and feeds them to a pixel-shuffle layer to result into the HR image features.

Filter Size: A large receptive field is important for utilizing spatial inter-dependencies in the structures present in the LR image. Since receptive-field of the network increases with network depth and/or filter size, depth of the network plays an crucial role in the super-resolution performance, as is evident from the existing state-of-the-art. A few initial works [17] did utilize larger filter size (larger than 3×3) to compensate for smaller depth, but showed only limited super-resolving performance. This can be attributed to the superior speed and parametric efficiency of a 3×3 filters above higher-size kernels. A stack of 3×3 filters is capable of learning more complex mapping via the non-linearity and the composition of abstraction. Hence, we use 3×3 filters for all the layers in our network.

Differences with [52],[54]: While [52] and [54] are proposed for a high-level computer vision task (object recognition), MDCN is suited for image restoration tasks and adopts the concept of blending a variety of dense topological connections at both macro and micro level. Other ncessary design changes include removal of the batch-normalization (BN) layers since they consume the same amount of GPU memory as convolutional layers, increase computational complexity, and hinder performance for image super-resolution. We also remove the pooling layers, since they could discard pixel-level information necessary for super-resolution. To enable higher growth rate, each MDCB contains with a 1×1 convolution layer, which fuses the information from the large number of features channels into a fewer feature channels and feeds these to the next MDCB block. Further each MDCB contains a third connection in the form of an additive operation between its input features and the output features.

3.3. Scale-Recurrent Design

Most existing SR algorithms treat super-resolution of different scale factors as independent problems without considering and utilizing mutual relationships among different scales in SR. Examples include EDSR [6], which

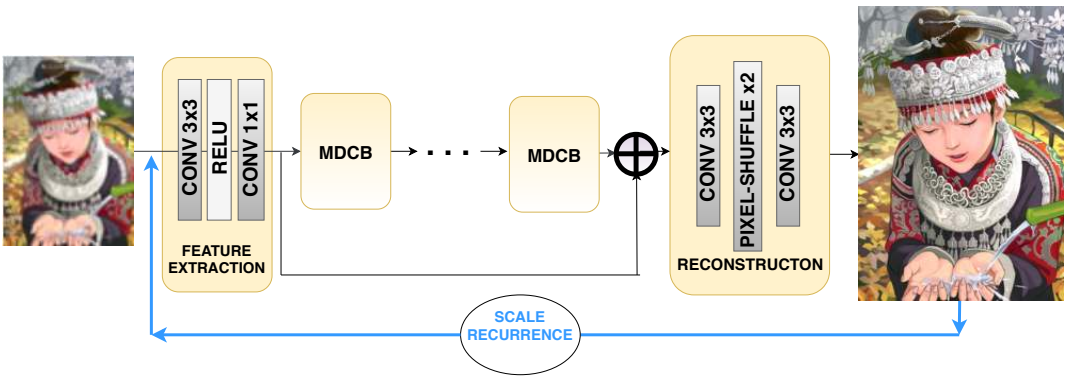


Figure 4: Scale-Recurrent utilization of our Mixed Dense Connection Network (MDCN). Output of 2× network is fed recursively for 4× and 8× super-resolution.

require many scale-specific networks that need to be trained independently to deal with various scales. On the other hand, architectures like VDSR [8] can handle super-resolution of several scales jointly in the single network. Training the VDSR model with multiple scales boosts the performance and outperforms scale-specific training, implying the redundancy among scale-specific models. Nonetheless, VDSR style architecture takes bicubic interpolated image as the input, which contains less information (as compared to the true LR image) leading to sub-optimal feature extraction; and performs convolutions in HR space, leading to higher computation time and memory requirements.

We propose a scale-recurrent training scheme which has the advantages of the approaches mentioned above, while subduing their limitations. Our networks global design is a multi-scale pyramid which recursively uses the same convolutional filters across the scales, motivated from the fact that a network capable of super-resolving an image by a factor of 2 can be recursively used to super-resolve the image by a factor 2^s, $s = 2, 3, \cdots$ Even with the same training data, the recurrent exploitation of shared weights works in a way similar to using data multiple times to learn parameters, which actually amounts to data augmentation regarding scales. We design the network to reconstruct HR images in intermediate steps by progressively performing a 2× up-sampling of the input from the previous level. Specifically, we first train a network to perform SR by a factor of 2 and then re-utilize the same weights to take the output of 2× as input and result into a output at resolution 4×. This architecture is then fine-tuned to perform 4× SR. We experimentally found that such initialization (training for task of 2× SR) leads to better convergence of the networks for larger scale factor. Ours is the first approach that efficiently re-utilizes the parameters across scales, which significantly reduces the number of trainable parameters while yielding performance gains for higher scale factors. An illustration of this scheme is shown in Fig. 4.

4. Perceptual and Objective Quality Trade-off

A recent study of various loss functions [9] has revealed that an algorithm can be potentially improved only in terms of its distortion (pixel-reconstruction error) or in terms of its perceptual quality (visual sharpness), but one at the expense of the other. This complementary property can be observed in the performance of two networks: SRGAN and SRResNet, proposed in [5], that utilize L2 loss and adversarial loss functions, respectively, to obtain

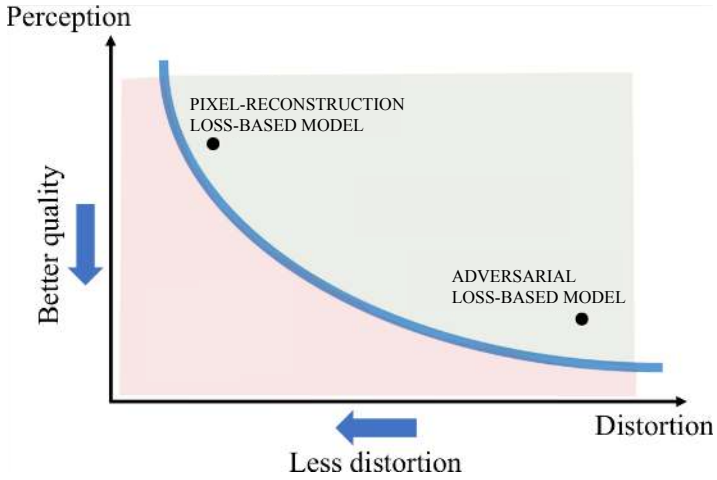


Figure 5: Perception-distortion trade-off: Left point represents the expected result when the network is trained with pixel-reconstruction loss; right point presents the expected result, when the network is trained using adversarial loss and deep-feature loss.

different type of super-resolution results. Interestingly, it was found that losses based on feature maps of a deep CNN eg. VGG-Net have higher correlation with both the objective quality as well as perceptual quality.

Most of the existing methods lack the flexibility of traversing from a better objective (MSE-based) quality to a better perceptual quality during test time, as the conventional loss functions do not support both of the quality measurements simultaneously. We propose to use two networks to overcome this issue. Our first network is trained with a weighted combination of L1 and VGG54 (on feature outputs of conv5-4 layer of VGG-Net) losses, so that it results into outputs with better objective quality. Motivated from SRGAN [5], our second network has the same architecture as the first one and it is trained with a combination of VGG loss and adversarial loss. The adversarial loss pushes our solution to the natural image manifold using a discriminator network that is trained to differentiate between the super-resolved images and original real HR images. The loss functions are described next.

Let θ represent the weights and biases in the network $\theta = W, B$. Given a set of training image pairs ${I_k}^L, {I_k}^H$, we minimize the following Mean Absolute Error (MAE):

$$l_{MAE}(\theta) = 1/N \sum_{k=1}^{N} \|F({I_k}^L, \theta) - {I_k}^H\|^2 \tag{6}$$

For perceptually superior results (having photo-realistic appearance), while not sacrificing the objective performance heavily, we also add the following VGG-based loss to our training objective:

$$l_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}({I_k}^H)_{x,y} - \phi_{i,j}(F({I_k}^L, \theta))_{x,y})^2 \tag{7}$$

Here $W_{i,j}$ and $H_{i,j}$ describe the dimensions of the respective feature maps within the VGG network.

Beside the the image reconstruction losses, a conditional adversarial loss is adopted that encourages a sharper

texture in the images generated by the network. This corresponds to the following objective function:

$$l_{CGAN}(F, D) = \mathbf{E}[\log D(I_k{}^L, I_k{}^H)] + \mathbf{E}[\log(1 - D(I_k{}^L, F(I_k{}^L, \theta)))], \tag{8}$$

where $\mathbf{E}$ represents the expectation operation and D represent a discriminator network. We use the same discriminator network as [5].

Once the two networks are trained, we pass each test image through them separately. The outputs are expected to have complementary properties. SRRDN returns an HR image (I_{HR1}) which is as close as possible to the ground-truth (in terms of MSE). Further, we use self-ensemble technique to improve MSE. However, as explained in [9], such objectively superior output can be perceptually inferior. On the other hand, SRRDN-GAN leads to a perceptually superior image (I_{HR2}), while sacrificing the objective quality (in terms of PSNR). The produced results I_{HR1} & I_{HR2} can be effectively combined to preserve the sharpness features from I_{HR2}, while bringing the intensities closer to I_{HR1}.

We show quantitative and qualitative results of this analysis in section 5.3.

5. Experimental Results on Single Image SR

Here, we evaluate our approach on standard datasets for different scale-factors. After discussing different experimental settings, we illustrate the results of our network, and compare them quantitatively as well as qualitatively with state-of-the-art approaches. The section concludes by meeting with the perception-distortion trade-off.

5.1. Settings

We next indicate the experimental settings about datasets, degradation models, training settings, implementation details, and evaluation metrics.

- **Datasets and degradation model:** Following [56, 6, 7, 25], we use 800 training images from DIV2K dataset [56] as training set. For testing, we use five standard benchmark datasets: Set5 [57], Set14 [15], B100 [58], Urban100 [26], and Manga109 [59]. We have considered bicubic interpolation for generating LR images, corresponding to the HR examples.

- **Training settings and implementation details:** Network's number of filters are chosen to reduce the number of parameters while not sacrificing the performance super-resolution. We choose a small number of feature-channels as input to each MDCN $F = 64$ and a modest growth-rate of $K = 36$. Number of MDCBs in hte network is set to 12 and the number of Dual Link Units within each MDCB is 6. Data augmentation is performed on the 800 training images, which are randomly rotated by 90°, 180°, 270° and flipped horizontally. In each training batch, 16 LR color patches with the size of 32×32 are extracted and fed as inputs.

Our model is trained using ADAM optimizor [60] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The initial leaning rate is set to 10^{-4} and then decreases to half every 2×10^5 iterations of back-propagation. We use PyTorch [61] to implement our models with a Titan Xp GPU.

- **Evaluation metrics:** We have considered PSNR and SSIM [62] for comparing our approach with state-of-the-arts. Note that higher PSNR and SSIM values indicate better quality. Following existing methods, we first converted the image from RGB color space to YCbCr color space, and then calculated the quality assessment metrics using the Y channel (luminance component). This is done by considering the significance of luminance component in visual perception of a scene as compared to the chromatic counterparts. For SR by factor s , we crop s pixels near image boundary before evaluation as in [6]. Some of the existing networks such as SRCNN [17], FSRCNN [18], VDSR [8], and EDSR [6] did not perform 8× super-resolution. To this end, we retrained the existing networks by using authors code with the recommended parameters.

  In subsection 5.3, we present perceptual and MSE score analysis of our networks to demonstrate the perceptual-objective quality trade-off.

- **Comparisons:** We considered bicubic interpolation technique and nine deep-learning based approaches for comparisons. These are SRCNN [17], FSRCNN [18], VDSR [8], LapSRN [21], MemNet [24], EDSR [6], SRMDNF [25], D-DBPN [63] and RDN [7]. For perceptual and objective super-resolution, we have performed qualitative comparisons with SRResNet,SRGAN [5] and ENet-E,Enet-PAT [23] in subsection 5.3. We have also encompassed the geometric self-ensemble strategy in testing our network for improvement, as has been done in [6, 7].

5.2. Results on Standard Benchmarks

The quantitative comparisons on the test sets (Set5, Set14, B100, Urban100, & Manga109) in terms of average PSNR & SSIM are given in Table 1 for scale factors ×2, ×3, ×4, and ×8. The results of our network and geometric self-ensemble strategy are shown in MDCN & MDCN+ rows, respectively. One can note that our MDCN strategy is able to out-perform most of the approaches for lower scale factors 2 & 3. Whereas, for higher factors such as 4 & 8, proposed MDCN+ is able to surpass all the state-of-the-art approaches. Observe that even without the geometric self-ensemble strategy, our MDCN is able to out-perform the existing approaches for higher scale factors.

However, for the scaling factor 2, the method of RDN shows close performance to our network, which can be attributed to their sheer number of parameters (120% more than our network). When the scaling factor becomes larger (e.g.,×3 and ×4), RDN does not hold the similar advantage over our network. In terms of parametric efficiency, we outperform all the methods with a large margin. Parametrically, MDSR is closest to ours, but leads to a lower performance. The parametric efficiency of MDSR over existing approaches like EDSR,RDN and DDBPN can be attributed to their training configuration. First, MDSR utilizes multi-scale inputs as VDSR does [8], which

Table 1: Quantitative results with bi-cubic degradation model. Best results are **highlighted**.

Method	Training Set	Scale	Set5		Set14		B100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	-	×2	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	30.80	0.9339
SRCNN [17]	91+ImageNet	×2	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.50	0.8946	35.60	0.9663
FSRCNN [18]	291	×2	37.05	0.9560	32.66	0.9090	31.53	0.8920	29.88	0.9020	36.67	0.9710
VDSR [8]	291	×2	37.53	0.9590	33.05	0.9130	31.90	0.8960	30.77	0.9140	37.22	0.9750
LapSRN [21]	291	×2	37.52	0.9591	33.08	0.9130	31.08	0.8950	30.41	0.9101	37.27	0.9740
MemNet [24]	291	×2	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	37.72	0.9740
EDSR [6]	DIV2K	×2	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
SRMDNF [25]	DIV2K+WED	×2	37.79	0.9601	33.32	0.9159	32.05	0.8985	31.33	0.9204	38.07	0.9761
D-DBPN [63]	DIV2K+Flickr	×2	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775
RDN [7]	DIV2K	×2	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	39.18	0.9780
MDCN (ours)	DIV2K	×2	38.24	0.9613	33.93	0.9207	32.34	0.9017	32.83	0.9353	39.09	0.9777
MDCN+ (ours)	DIV2K	×2	**38.30**	**0.9616**	**34.05**	**0.9217**	**32.39**	**0.9023**	**33.05**	**0.9369**	**39.32**	**0.9783**
Bicubic	-	×3	30.39	0.8682	27.55	0.7742	27.21	0.7385	24.46	0.7349	26.95	0.8556
SRCNN [17]	91+ImageNet	×3	32.75	0.9090	29.30	0.8215	28.41	0.7863	26.24	0.7989	30.48	0.9117
FSRCNN [18]	291	×3	33.18	0.9140	29.37	0.8240	28.53	0.7910	26.43	0.8080	31.10	0.9210
VDSR [8]	291	×3	33.67	0.9210	29.78	0.8320	28.83	0.7990	27.14	0.8290	32.01	0.9340
LapSRN [21]	291	×3	33.82	0.9227	29.87	0.8320	28.82	0.7980	27.07	0.8280	32.21	0.9350
MemNet [24]	291	×3	34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376	32.51	0.9369
EDSR [6]	DIV2K	×3	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17	0.9476
SRMDNF [25]	DIV2K+WED	×3	34.12	0.9254	30.04	0.8382	28.97	0.8025	27.57	0.8398	33.00	0.9403
RDN [7]	DIV2K	×3	34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653	34.13	0.9484
MDCN (ours)	DIV2K	×3	34.71	0.9295	30.58	0.8473	29.28	0.8096	28.75	0.8656	34.18	0.9485
MDCN+ (ours)	DIV2K	×3	**34.82**	**0.9303**	**30.68**	**0.8486**	**29.34**	**0.8106**	**28.96**	**0.8686**	**34.50**	**0.9500**
Bicubic	-	×4	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SRCNN [17]	91+ImageNet	×4	30.48	0.8628	27.50	0.7513	26.90	0.7101	24.52	0.7221	27.58	0.8555
FSRCNN [18]	291	×4	30.72	0.8660	27.61	0.7550	26.98	0.7150	24.62	0.7280	27.90	0.8610
VDSR [8]	291	×4	31.35	0.8830	28.02	0.7680	27.29	0.0726	25.18	0.7540	28.83	0.8870
LapSRN [21]	291	×4	31.54	0.8850	28.19	0.7720	27.32	0.7270	25.21	0.7560	29.09	0.8900
MemNet [24]	291	×4	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	29.42	0.8942
EDSR [6]	DIV2K	×4	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148
SRMDNF [25]	DIV2K+WED	×4	31.96	0.8925	28.35	0.7787	27.49	0.7337	25.68	0.7731	30.09	0.9024
D-DBPN [63]	DIV2K+Flickr	×4	32.47	0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137
RDN [7]	DIV2K	×4	32.47	0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028	31.00	0.9151
MDCN (ours)	DIV2K	×4	32.59	0.8994	28.84	0.7877	27.73	0.7416	26.62	0.8030	31.03	0.9160
MDCN+ (ours)	DIV2K	×4	**32.66**	**0.9003**	**28.92**	**0.7893**	**27.79**	**0.7428**	**26.785**	**0.8065**	**31.34**	**0.9184**
Bicubic	-	×8	24.40	0.6580	23.10	0.5660	23.67	0.5480	20.74	0.5160	21.47	0.6500
SRCNN [17]	91+ImageNet	×8	25.33	0.6900	23.76	0.5910	24.13	0.5660	21.29	0.5440	22.46	0.6950
FSRCNN [18]	291	×8	20.13	0.5520	19.75	0.4820	24.21	0.5680	21.32	0.5380	22.39	0.6730
SCN [19]	91	×8	25.59	0.7071	24.02	0.6028	24.30	0.5698	21.52	0.5571	22.68	0.6963
VDSR [8]	291	×8	25.93	0.7240	24.26	0.6140	24.49	0.5830	21.70	0.5710	23.16	0.7250
LapSRN [21]	291	×8	26.15	0.7380	24.35	0.6200	24.54	0.5860	21.81	0.5810	23.39	0.7350
MemNet [24]	291	×8	26.16	0.7414	24.38	0.6199	24.58	0.5842	21.89	0.5825	23.56	0.7387
MSLapSRN [22]	291	×8	26.34	0.7558	24.57	0.6273	24.65	0.5895	22.06	0.5963	23.90	0.7564
EDSR [6]	DIV2K	×8	26.96	0.7762	24.91	0.6420	24.81	0.5985	22.51	0.6221	24.69	0.7841
D-DBPN [63]	DIV2K+Flickr	×8	27.21	0.7840	25.13	0.6480	24.88	0.6010	22.73	0.6312	25.14	0.7987
MDCN (ours)	DIV2K	×8	27.32	0.7867	25.13	0.6472	24.93	0.6023	22.84	0.6370	25.00	0.7956
MDCN+ (ours)	DIV2K	×8	**27.39**	**0.7895**	**25.25**	**0.6499**	**24.99**	**0.6039**	**23.02**	**0.6422**	**25.26**	**0.8009**

enables them to estimate and fuse different scale features. Second, MDSR uses larger input patch size (48 against 32) for training. As most images in Urban100 contain self-similar structures, larger input patch size for training allows a very deep network to grasp more information by using large receptive field better. However, in our MDCN, we do not use multi-scale information or larger patch size. Moreover, our MDCN+ can achieve further improvement with the geometric self-ensemble technique. This is due to our effective combination of residual and dense connections through DBCN block with scale-recurrent strategy.

The improvement of results can be observed visually in Figs. 6, 7 & 8 for scale factors 3, 4 and 8, respectively. In Fig. 6 (img_033), one can observe that most of the approaches produce smoother results as compared to ours.

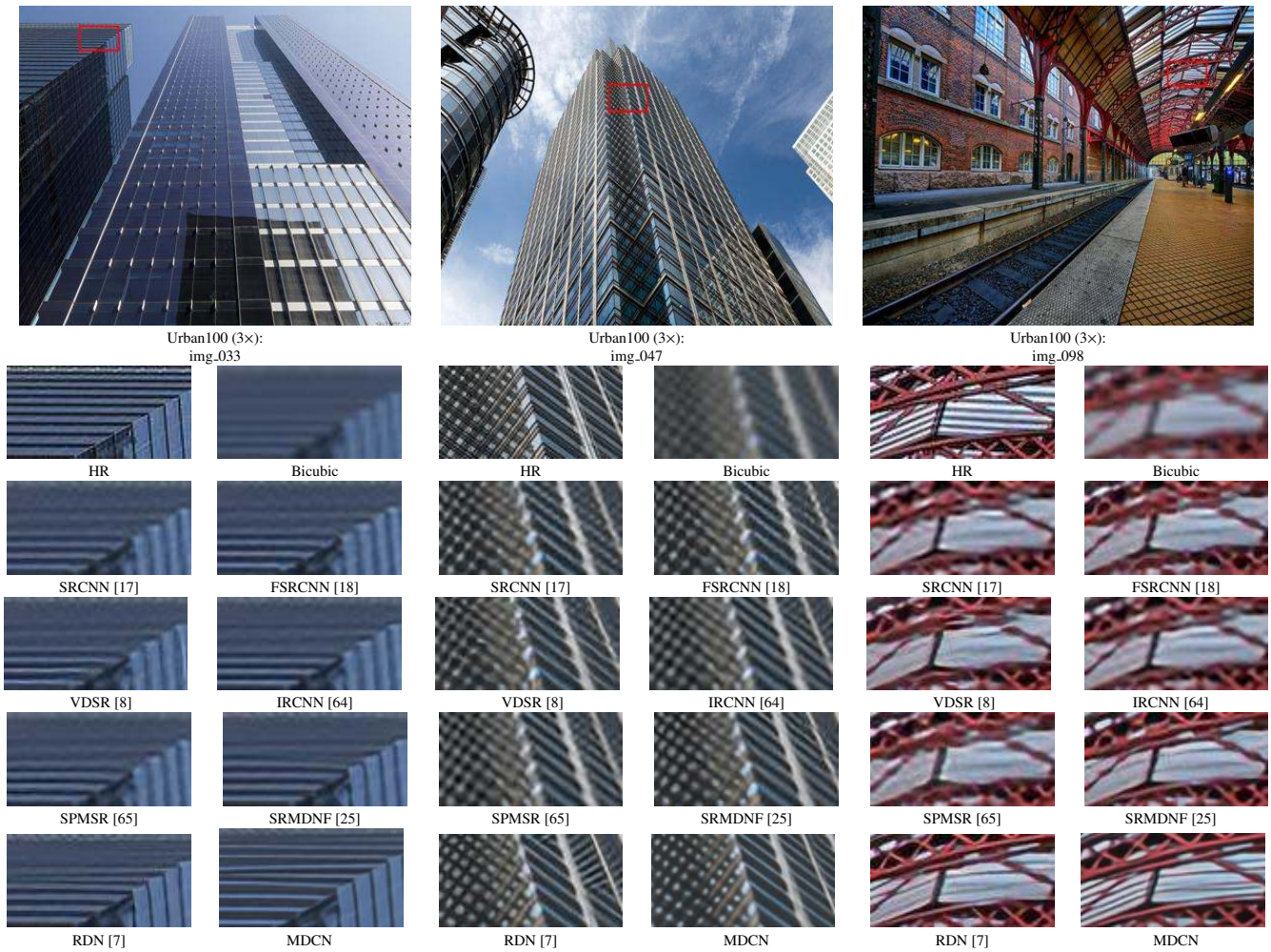


Figure 6: Visual comparison for 3× SR with bi-cubic degradation on Urban100 dataset.

Whereas, for img_037, the approaches yield results, where some of the linear structures are disorientated. However, our MDCN is able to produce result with appropriate line structure. On the other hand, most of the existing methods fail to reproduce the line structures of the roof of the railway station in img_098, whereas our strategy helps in bringing back those details in the result.

For scale factor 4, the size of the input image itself is too small to provide different details to the network. Hence, the existing approaches fail to maintain different structures faithfully. For example, in img_004 of Fig. 7, the elliptical shape structure is completely demolished and other regions appear to be parallel structured vertical lines. However, our MDCN is able to maintain somewhat similar structure to the ground-truth. The img_092 consists of lines mainly in two directions. However, the existing models are not able to maintain the line orientation as per with the ground truth. Moreover, some of the approaches creates blocky artifacts due to generation of some unwanted lines. In contrast, our MDCN is able to maintain the line orientations appropriately. Blurring artifacts are quite evident for such larger scale factors, as can be observed in third scene of Fig. 7 for the results of most of the existing approaches. The result of our model has less affected by blur.

Such blurring artifact increases with larger scale factor such as 8. This is because for very high scale factor, the

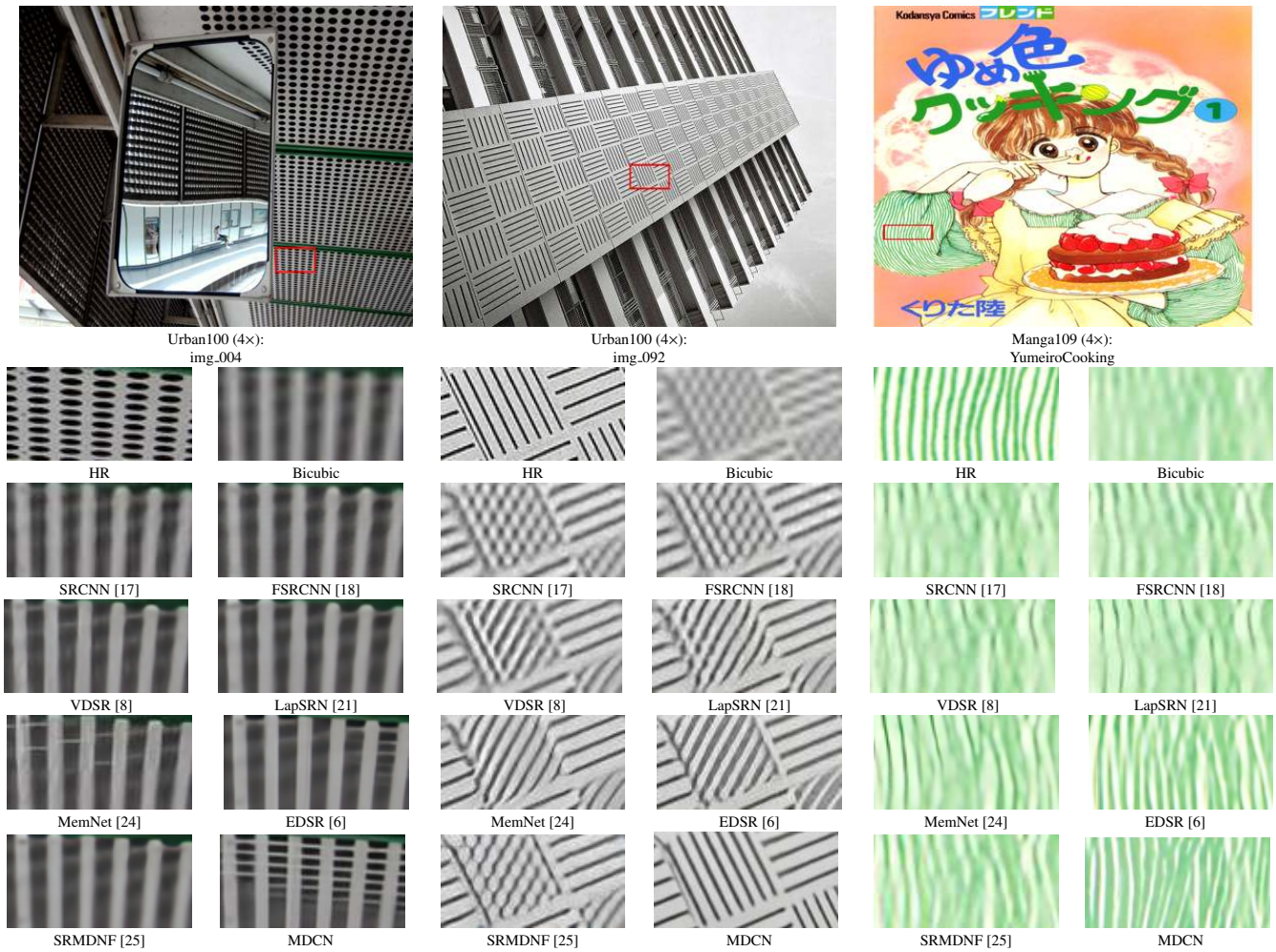


Figure 7: Visual comparison for 4× SR with bi-cubic degradation on Urban100 and Manga109 datasets.

resolution of the input image is quite low with minimum details. Thus, super-resolving these LR images by a factor of 8 is very challenging. As a consequence, most of the approaches suffer from over-smoothing effect, as can be seen in Fig. 8. Among the existing methods, EDSR [6] can bring out some texture details but the orientations of those are not appropriate. On contrary, our MDCN is able to bring out texture details with appropriate orientation. This is due to our elegant scale-recurrent framework, which enables us to carry-over the improvements for lower scale factors via weight transfer strategy.

5.3. Meeting the perception distortion trade-off

As explained in Section 4, we also analyzed the effect of different loss configurations on the performance of the network for single image super-resolution. Our networks were trained for factor 4 with those loss functions (mentioned in Section 4) on the DIV2K train dataset (800 images), and tested on the DIV2K validation dataset (100 images). Two metrics were used for evaluation, one of them is mean squared error (MSE), and the other is a perceptual metric, which is estimated as [66]

$$P(I) = \frac{1}{2}((10 - M(I)) + N(I). \tag{9}$$

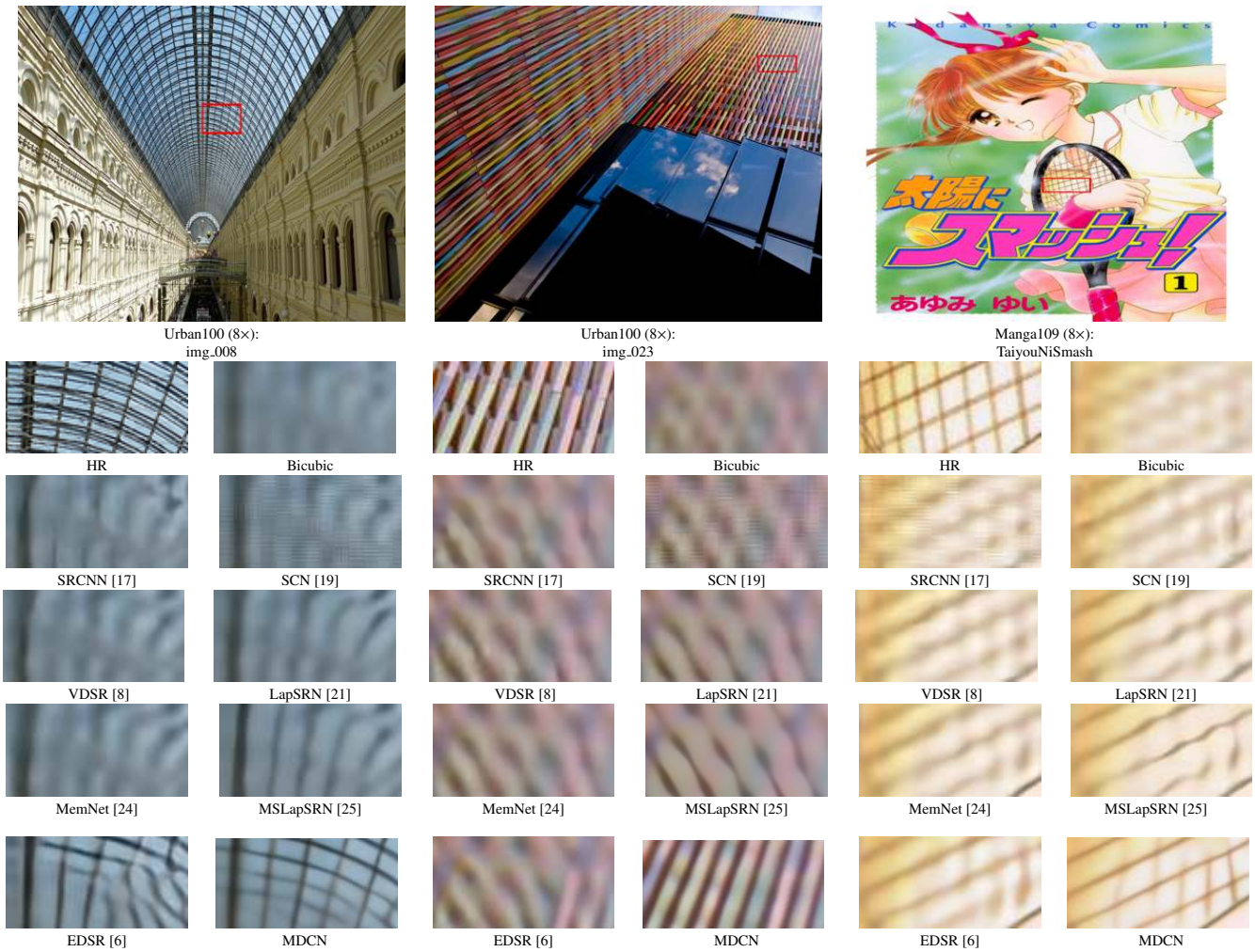


Figure 8: Visual comparison for 8× SR with bi-cubic degradation on Urban100 & Manga109 datasets.

Here $M(I)$ and $N(I)$ are computed using [67] and [68], respectively. In both the metrics, lower value depicts better result. The perceptual score is plotted against MSE in Fig. 9. The points labeled in blue represent loss configurations which gave higher weights to reconstruction quality (lower MSE). Specifically, we trained our network using a weighted combination of L1 loss and VGG loss (eq. 7). The slight variation in the performance is due to small differences in the duration of training as well as the relative coefficient of VGG loss. The points labeled in red represent loss configurations which gave higher weight to perceptual quality (lower MSE). Specifically, we trained our network using a weighted combination of VGG loss and conditional GAN loss (eq. 8). The slight variation in the performance is due to small differences in the duration of training as well as the relative coefficient of adversarial loss.

Note that the distribution of these evaluations follows the curve explained by [9]. Specifically, the point at the left extreme corresponds to the network purely trained using L1 loss from scratch. Consistent with the findings of[9], it leads to the lowest MSE but a very poor perceptual score. On the other hand, the right-most point corresponds to a network fine-tuned using very high weight for the adversarial loss (no L1 loss). This yields one of the best perceptual performance but performs poorly in terms of MSE. Our results show strong agreement with the

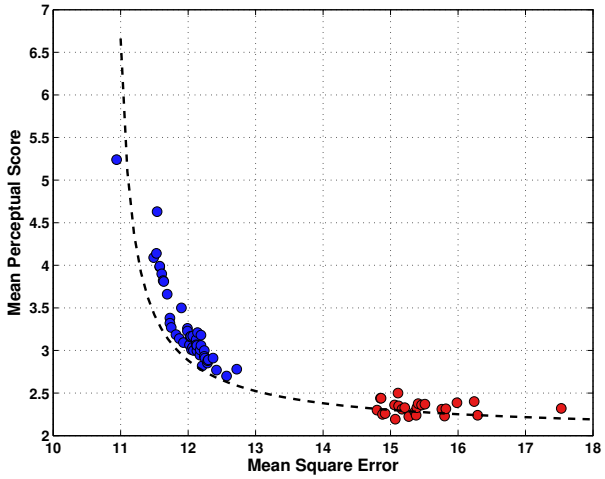


Figure 9: Perceptual score vs mean squared error for scale factor 4 on DIV2K validation dataset. Different points are produced by using different weights to the loss functions.

argument that an algorithm can be potentially improved only in terms of its distortion or in terms of its perceptual quality, but one at the expense of the other. These results can be combined to produce a balanced result. A few super-resolved outputs corresponding to this experiment are shown in Fig. 10, where the perceptually superior results (MDCN-P) are produced by using $(5 * l_{VGG} + 0.25 * l_{CGAN})$ loss configuration. One can observe that our

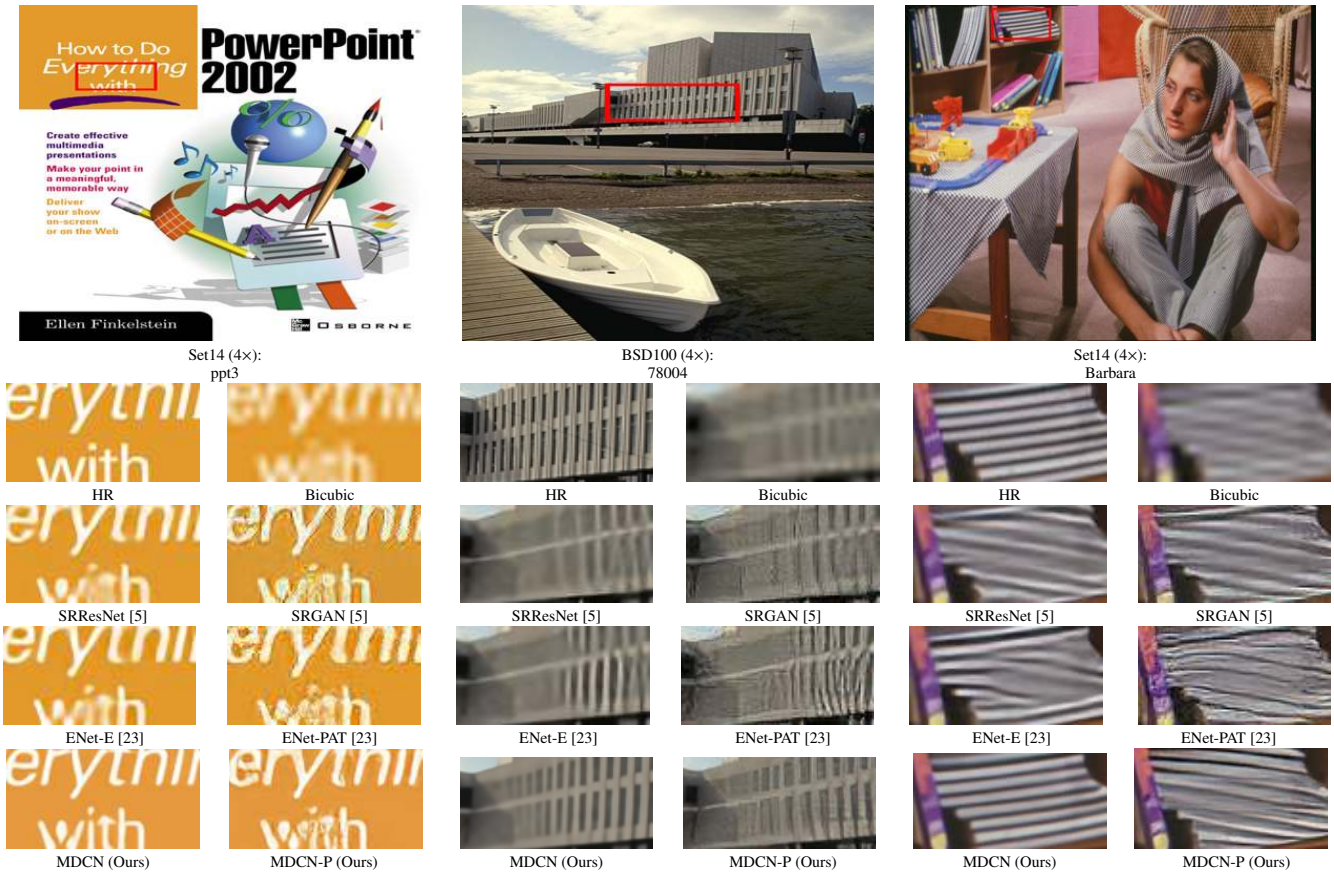


Figure 10: Visual comparison for 4× SR on images from Set14 and BSD100 datasets.

network (MDCN) is able to outperform existing approaches (SRResNet [5] and ENet-E [23]) since our results are

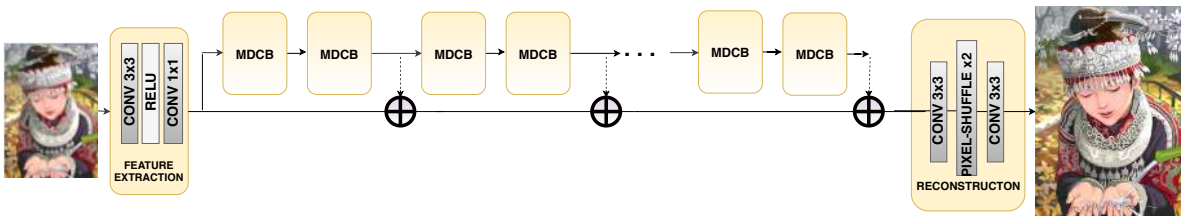


Figure 11: Variable depth supervision for resource-aware SR. Results can be generated from different depth of our network using different number of blocks.

objectively closer to the ground truth. On the other hand, a combination of VGG loss and adversarial based loss helps our network (MDCN-P) to produce perceptually superior results, while partially preserving the image content as compared to the existing approaches of SRGAN [5] and ENet-PAT [23] (which contain artifacts).

6. Network Analysis

Here, we analyze and compare our networks in various aspects such as effect of scale recurrent design, depth variation, parameter size, and convergence curve.

6.1. Effect of scale-recurrent design

We study the effects of scale recurrent design. Note that ours is the first method to re-utilise the layer prameters across scales. Although methods like DRRN and VDSR have proposed a single network for different scale factors, their layer weights are utilised only once for a given image. However, we experimentally found that the weights learnt for smallest scale (e.g. $\times 2$) can be recursively used to achieve state-of-the-art SR performance at higher scale factors of $\times 4$ and $\times 8$, without any increment in network parameters. For comparison, we designed a network which follows the procedure used by EDSR for multiple scale training. Specifically, it replaces the reconstruction module (pconv-layer and pixel-shuffle layer) of x2 network with a reconstruction module for x4, keeping the rest of the network same. It can be inferred from Table 2 that the performance of our scale-recurrent network is superior to

Method	Set5	Set14	BSD100	Urban100	Managa
Baseline network	32.48	28.73	27.61	26.51	30.90
Scale-Recurrent Network	32.59	28.84	27.73	26.62	31.03

Table 2: Advantage of scale recurrent framework over the baseline one. The entries in the table are average PSNR values for different datasets for scale factor 4.

the baseline network.

6.2. Variable depth supervision for resource-aware super-resolution

We explore the ability of our network in super-resolving a scene by considering variable depths of it. The depth is quantified as the number of mixed dense connection blocks by keeping rest of the network intact. We have considered 2, 4, 6, 8, 10, 12, 14, and 16 blocks separately for training different depth versions of our model. The trained models are then used to evaluate the performance of the network for Set14 dataset for scale factor 2,

and the behavior in terms of average PSNR are plotted with respect to the number of parameters in Fig. 12. N1 to N8 represent the depths with number of blocks 2 to 16, respectively. For comparison, we have have included the performance of state-of-the-art approaches in the plot. One can observe that our MDCN with comparable number of parameters is able to produce better results than some approaches such as VDSR, LapSRN, MemNet. By increasing the depths with maximum 9 million parameters, our MDCN is able to outperform the best performing approaches such as RDN and EDSR, which have 22 and 43 million parameters. The performance of our network with less number of parameters illustrates the usefulness of it in limited computational resources.

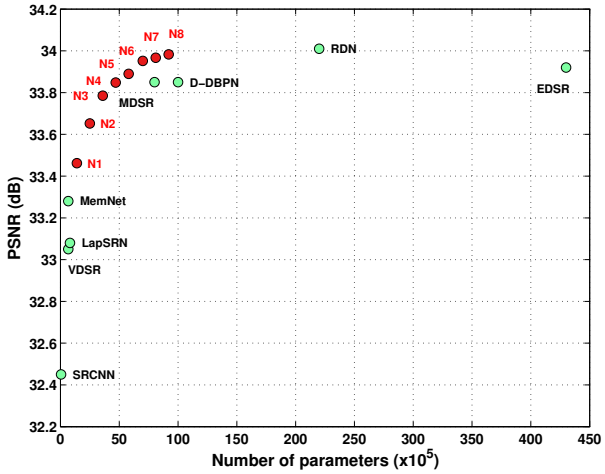


Figure 12: Performance of our network on Set14 dataset (x2) using variable depth versions along with the results of state-of-the-art approaches. Different depth versions of our network has different number of parameters. N1 to N8 depict our network with number of blocks 2, 4, 6, 8, 10, 12, 14, and 16, respectively.

6.3. Number of dual-link units

In this experiment, we vary the number of dual-link units in each MDCB, which is a key hyper-parameter of our super-resolution network. To find its optimal value, we have designed and trained 3 versions of the network: which are build using 5, 6, and 7 dual link units in the MDCBs, respectively. Fig. 13 shows comparisons of the convergence process of these 3 models and Table 3 compares their test performance. It can be observed that the training performance as well as the quantitative results get better with increase in the number of units, but saturates while increasing number to 7. We chose N=6 in our proposed model, since the improvement beyond 6 is marginal and it serves as a good balance between efficiency and performance.

Number of units	Set5	Set14	BSD100	Urban100	Manga109
5 units	38.27	33.90	32.37	32.90	39.19
6 units	38.30	34.05	32.39	33.05	39.32
7 units	38.30	34.07	32.41	33.13	39.40

Table 3: Effect of increasing the number of dual-link units in our network. The entries in the table are average PSNR values on different datasets for scale factor 2.

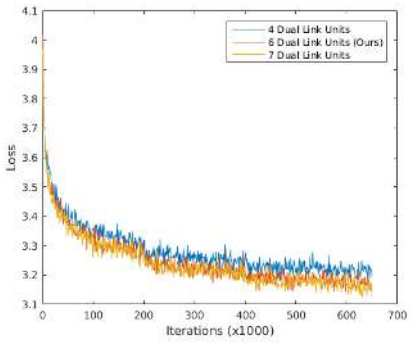


Figure 13: Comparison of networks with different number of dual-link units through their training performance over the same dataset.

6.4. Comparison of dual-link units with residual blocks or dense blocks

We experimentally analyzed the advantages of the proposed dual-link units over residual unit or dense units, by designing and training two baselines: MDCN without dense connections and MDCN without residual connections. For fair comparison, we have constructed these networks with about the same number of parameters as our proposed network. Specifically, for the model with no residual connections, we increase the growth rate (K) from 36 to 52. For the model with no dense connections, we increase the number of feature-channels as input to each MDCN (F) to 152.

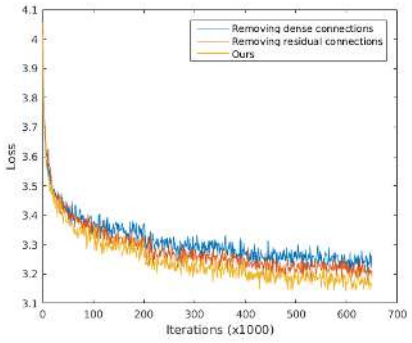


Figure 14: Comparison of our network with residual and dense baselines through training performance over the same dataset.

Connection type	Set5	Set14	BSD100	Urban100	Manga109
No dense connections	38.22	33.95	32.33	32.66	38.98
No residual connections	38.28	33.98	32.35	32.85	39.18
Ours	38.30	34.05	32.39	33.05	39.32

Table 4: Test performance comparison of our dual-link based network with residual and dense baselines. The values are average PSNR scores for different datasets for scale factor 2.

Fig. 14 shows comparisons of the convergence plots of these 3 models and Table 4 compares their test performance on standard datasets. We find that only including dense connections is better than only residual connections for the same network size. This finding is in agreement with the literature that tests densely connected networks over residual ones on other vision problems [51]. Importantly, it can be observed that the training performance as

well as the quantitative test results get better by carefully including both connections into the network design. In other words, removing any of the two links decreases the performance on training as well as the testing data.

6.5. Ordering of addition and concatenation connections

The Dual Link Unit utilizes the strengths of both residual and dense connections while addressing their limitations and is a key component of the MDCN. With this motivation, an alternative (second configuration) to our dual-link unit is a unit that performs the concatenation step before additive operation. We compare the two choices below.

If the feature-map which is fed as input to dual link unit has c number of channels, the output feature map will have $c + K$ channels. Although the concatenation is always performed at the end of the feature map, there are exponentially many combinations to pose the additive modules' positions along multiple layers. Learning mechanism for such variable positioning is currently unavailable since their arrangement is not derivable directly. Therefore, we choose the positioning technique of *shifted addition* as shown in Fig. 2. Specifically, the position of additive part (denoted by red color) exactly aligns with the growing boundary of entire feature embedding when the concatenated parts (denoted by green color) increase the overall feature dimension. Based on the above understanding, we can investigate the feature redundancy and feature exploration capability of the two configurations.

Feature Redundancy: Following our choice of addition before concatenation (first configuration), only $c - F$ number of channels remain unaltered between the input and output. The rest of the channels in output will be either new or modified features. However, if concatenation is performed before addition (second configuration), $c + K - F$ number of channels stay unaltered. It can be seen that the second configuration leads to higher number of unmodified features being propagated to future layers. This is suboptimal since one of the goals of dual link units is to minimize the feature redundancy.

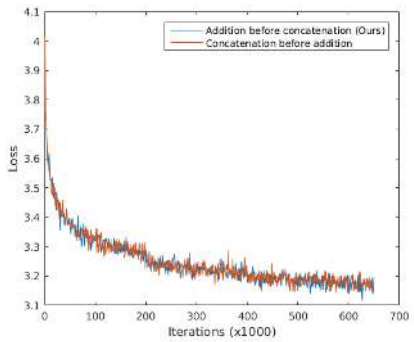


Figure 15: Comparison of networks through their training performance over the same dataset.

Feature Exploration: The second configuration has the advantage of facilitating repeated modifications of a few channels in the feature-map. Specifically, K channels introduced by the concatenation operation also undergo update due to the upcoming additive module. Since this is not the case in our design, none of the channels undergo

Method	Set5	Set14	BSD100	Urban100	Managa
Concatenation before addition	38.28	34.04	32.36	33.05	39.33
Our configuration	38.30	34.05	32.39	33.05	39.32

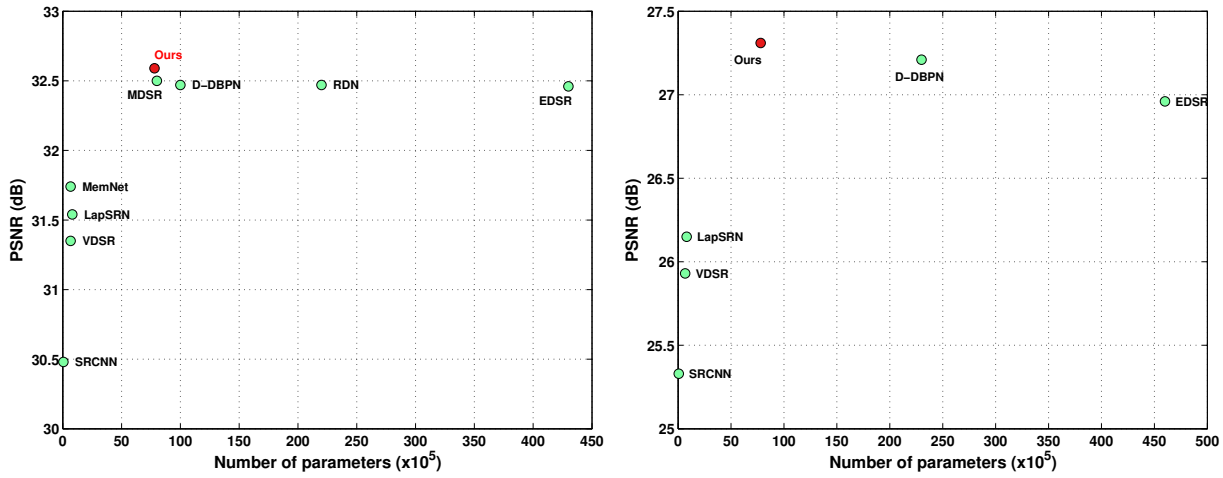
Table 5: Comparison of two configurations for designing dual-link units. The entries in the table are average PSNR values for different datasets for scale factor 2.

repeated modification. Repeated update of selected channels can lead to learning of more complex features and hence the second configuration is better in this aspect.

In principle, an effective configuration is one that leads to maximum amount of feature modification across the layers, while minimizing the number of learn-able parameters. This requirement is satisfied by both the configurations due to their different advantages.

We experimentally compare the two choices by training the modified network (which uses the second configuration). Fig. 15 shows the training curves corresponding to our network and the modified network. It can be seen that our network's score is equal or slightly better than the modified network and similar behavior is reflected in test scores in Table 5. Note that the number of parameters is exactly the same in two networks and hence the improvement is purely a result of the practical choice. This difference can be attributed to lower feature redundancy in our configuration, and is the reason why we choose to perform addition before concatenation in our dual-link units.

6.6. Model size



(a) Results on Set5 (4×) (b) Results on Set5 (8×)

Figure 16: Performance of existing approaches along with ours against the number of parameters. Results are evaluated on Set5.

We compare the ability of our network parametrically with state-of-the-art approaches for scale factors 4 & 8 in Fig. 16. Some of the approaches such as SRCNN, VDSR, LapSRN and MemNet require less number of parameters but produce inferior results than ours. Note that our MDCN is able to produce better results with less number of parameters as compared to the best-performing approaches (D-DBPN, RDN, EDSR) for both the scale factors.

This parametric efficient property is one of the main advantages of our MDCN over the existing approaches.

6.7. Convergence curve

The advantage of mixed-dense connections manifests itself in form of significant improvement in propagation of error gradients during training. This can be seen in Fig. 17, where we depict the convergence curve of our MDCN for scale factor 2 along with different lines, representing average PSNR values of existing approaches.

The proposed architecture's efficiency is demonstrated in the fact that its performance exceeds notable super-resolution baselines within just 2×10^4 steps of training.

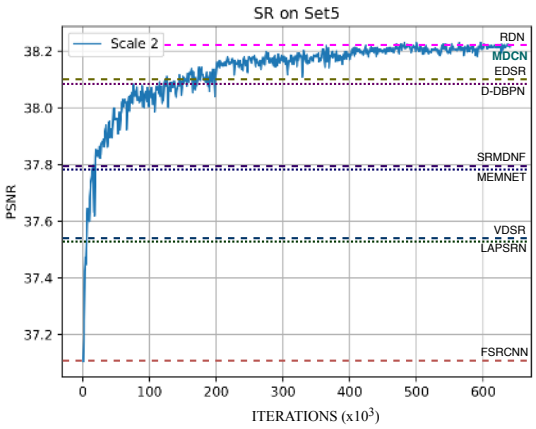


Figure 17: Convergence curve of our network over iterations along with the final performance of state-of-the-art approaches. Results are evaluated on Set5 for scale factor 2.

7. Video Super-Resolution

We extend our MDCN for video SR through a multi-image restoration approach. Although our trained SISR network can be directly utilized for the video SR task by processing each frame of the video individually, access to multiple neighboring LR frames can fundamentally reduce the ill-posedness of the SR task and potentially yield higher reconstruction quality.

We experimentally verified this fact by training a modified version of our network that accepts 5 consecutive LR frames (includes the frame to super-resolve and its 4 neighbors) as input. The frames are concatenated along the channel dimension before being fed to the first layer our network, which is modified to accept 15 channel inputs. This is referred to as early fusion technique in video SR literature.

The source for training data for this task was the Vimeo Super-Resolution (VSR) dataset [69] that contains 64612 training samples. Each sample in the dataset contains seven consecutive frames with 448×256 resolution. We addressed the task of $\times 4$ SR, where our network can reap the joint benefits of MDCBs and scale-recurrence. We used batch size of 8, where each sample contained 5 LR frames of 32×32 resolution and 1 HR frame of size 128×128. Our network was trained using Adam optimizer with a learning rate of 10^{-4} for 4×10^4 iterations.

7.1. Experimental results on video SR

For evaluation, we have considered standard Vid4 dataset, which consists of 4 videos. The results are compared with state-of-the-art video SR approaches such as VSRnet [44], Bayesian [40], ESPCN [70], VESPCN [45], and Temp_robust [71]. The quality assessment metrics are computed using the results provided by the respective authors of the approaches. The metrics for each of the approaches have been computed by considering 30 frames and by removing 8 boundary pixels from each of them.

Metric	Bicubic	Bayesian	VSRnet	ESPCN	VESPCN	Ours
PSNR	25.38	25.64	26.64	26.97	27.25	**27.55**
SSIM	0.7613	0.8000	0.8238	0.8364	**0.8400**	0.8332

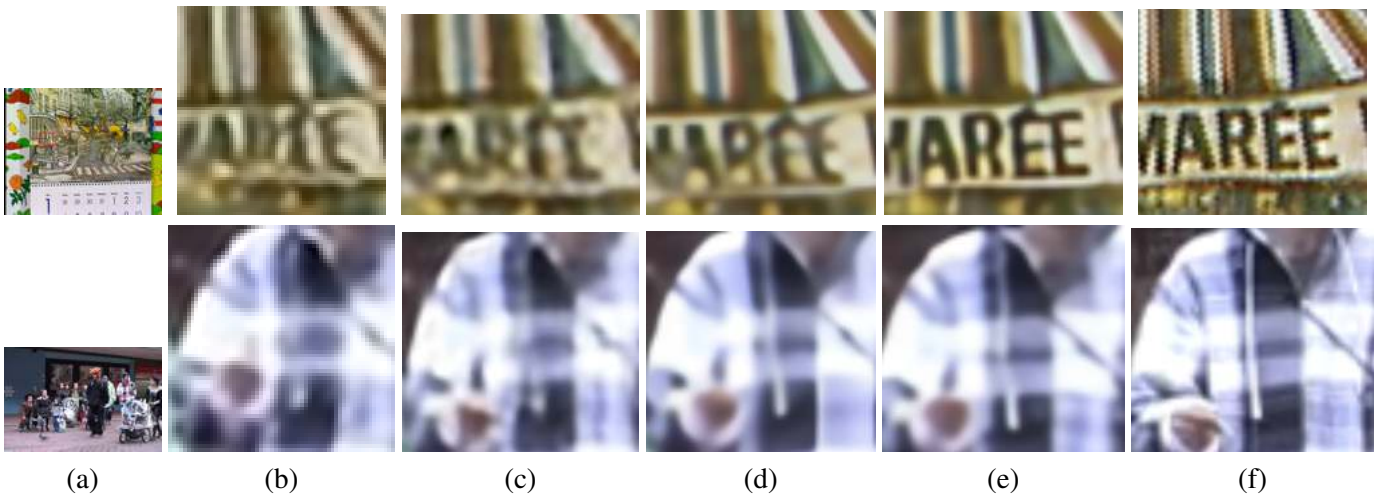
Table 6: Comparison of video SR results on the standard Vid4 dataset for scale factor 3. Best results are **highlighted**.

We have provided average PSNR and SSIM scores of different approaches along with our SISR model in Table 6 for scale factor 3. The comparison shows that our SISR network is able to out-perform the competing videoSR approaches in terms of PSNR for scale factor 3. Next, we evaluate our models on the challenging task of $\times 4$ video SR. Average PSNR and SSIM score for each video for scale factor 4 on the same dataset are kept in Table 7. One can observe that for scale factor 4, our MDCN is able to out-perform most of the video SR approaches and produce comparable results to Temp_robust [71]. This improvement can also be observed visually in Fig. 18(d) for scale factor 4.

Video	Metric	Bi-cubic	VSRnet	ESPCN	VESPCN	Temp_robust	Ours	Ours(multi-frame)
Calendar	PSNR	20.51	21.27	21.69	21.92	22.14	22.38	**23.36**
	SSIM	0.5622	0.6390	0.6736	0.6858	0.7052	0.7198	**0.7786**
City	PSNR	25.04	25.59	25.80	26.15	26.31	26.02	**27.20**
	SSIM	0.5969	0.6520	0.6687	0.6947	0.7218	0.6926	**0.7818**
Foliage	PSNR	23.62	24.44	24.62	24.95	25.07	24.79	**25.86**
	SSIM	0.5689	0.6451	0.6522	0.6731	0.7002	0.6639	**0.7386**
Walk	PSNR	25.97	27.49	27.99	28.21	28.05	28.42	**29.55**
	SSIM	0.7957	0.8432	0.8584	0.8594	0.8583	0.8705	**0.8940**
Average	PSNR	23.78	24.70	25.02	25.31	25.39	25.40	**26.49**
	SSIM	0.6309	0.6948	0.7132	0.7282	0.7464	0.7367	**0.7982**

Table 7: Comparison of PSNR values on the standard Vid4 dataset for scale factor 4. Best results are **highlighted**.

For higher scale factors e.g. 4, SISR becomes highly ill-posed and availability of neighboring LR frames to the network becomes crucial. Hence, we have provided qualitative comparisons between prior works, our SISR network and our multi-frame based network for $\times 4$ video SR task in Fig. 18. As can be observed from the *calendar* example (first row), our multi-frame approach is able to produce the best results, wherein the readability of letters has further improved as compared to the existing methods and our single-frame based approach. For the frame of *walk* video, one can note that both versions of our network are able to maintain the sharpness of the vertical



(a) (b) (c) (d) (e) (f)

Figure 18: Comparisons for video SR (x4) on vid4 dataset: (a) LR frame, (b) VESPCN, (c) Temp_robust, (d) Our model:single-frame based, (d) Our model:multiple-frame based, and (f) ground truth. First row represents the results of a frame of *calendar* video, and second row depicts the results of a frame of *walk*.

(a) (b) (c) (d) (e) (f)

Figure 19: Comparison of output frames for video SR (x4) on a zoomed in region in the video *walk* from vid4 dataset. The subfigures contain 8 consecutive frames from the results of (a) bicubic interpolation, (b) VESPCN, (c) Temp_robust, (d) Our model (single-frame based), (d) Our model (multiple-frame based), and (f) ground truth. *Videos can be viewed by clicking on the images, when document is opened in Adobe Reader*.

line much better than the competing approaches. In addition, Fig. 19 shows 8 consecutive output frames for each method. (Please click on the figures to play the videos). The results of VESPCN and Temp_robust partially suffer from distorted edges, missing texture and temporal fluctuations. Our SISR model leads to sharper results with fewer distortions but is limited in its capability to maintain temporal smoothness. In contrast, The frames estimated by our multi-frame based network are sharper, contain minimum fluctuations and are qualitatively consistent with the ground-truth video. These improvements are quantitatively reflected in Table 7, where our method scores atleast 1 dB higher than all prior works, including our SISR model. Our network's superior performance demonstrates its capability of implicitly handing the temporal shifts and extracting HR information that is distributed across multiple LR frames.

8. Conclusions

We proposed a novel deep learning based approach for single image super-resolution. Our single-image SR network MDCN has been designed by effectively combining the residual and dense connections. The effective combination assists in better information flow through the network layers by reducing the redundant features and

gradient vanishing problem. The scale recurrent framework has yielded better performance for higher scale factors with less number of parameters. The robustness of the network has been demonstrated for different scale factors on different datasets. The conventional pixel level loss functions in the network fails to produce perceptually better results. We included VGG loss as well as GAN based loss to generate photo-realistic results. Different weights to those loses synthesized different results that follow the perception-distortion trade-off. Further, the proposed MDCN structure has been utilized for video SR as a multi-frame restoration model. The resultant videos demonstrated the capability of our network in producing spatio-temporally consistent HR frames.

References

[1] J. Yang, J. Wright, T. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Transactions on Image Processing 19 (11) (Nov. 2010) 2861–2873. doi:10.1109/TIP.2010.2050625.

[2] W. Dong, L. Zhang, G. Shi, X. Wu, Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, IEEE Transactions on Image Processing 20 (7) (2011) 1838 –1857. doi:10.1109/TIP.2011.2108306.

[3] S. Mandal, A. Bhavsar, A. K. Sao, Noise adaptive super-resolution from single image via non-local mean and sparse representation, Signal Processing 132 (2017) 134 – 149. doi:http://dx.doi.org/10.1016/j.sigpro.2016.09.017.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[5] C. Ledig, L. Theis, F. Huszr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 105–114. doi:10.1109/CVPR.2017.19.

[6] B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee, Enhanced deep residual networks for single image super-resolution, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1132–1140. doi:10.1109/CVPRW.2017.151.

[7] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[8] J. Kim, J. K. Lee, K. M. Lee, Accurate image super-resolution using very deep convolutional networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1646–1654. doi:10.1109/CVPR.2016.182.

[9] Y. Blau, T. Michaeli, The perception-distortion tradeoff, arXiv preprint arXiv:1711.06077.

[10] X. Zhang, E. Lam, E. Wu, K. Wong, Application of Tikhonov regularization to super-resolution reconstruction of brain MRI images, in: X. Gao, H. Mller, M. Loomes, R. Comley, S. Luo (Eds.), Medical Imaging and Informatics, Vol. 4987 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 51–56. doi:10.1007/978-3-540-79490-5_8.

[11] A. Marquina, S. J. Osher, Image super-resolution by TV-regularization and bregman iteration, Journal of Scientific Computing 37 (2008) 367–382. doi:10.1007/s10915-008-9214-8.

[12] A. Kanemura, S. ichi Maeda, S. Ishii, Superresolution with compound markov random fields via the variational {EM} algorithm, Neural Networks 22 (7) (2009) 1025 – 1034. doi:10.1016/j.neunet.2008.12.005.

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in: IEEE 12th International Conference on Computer Vision, 2009, pp. 2272 –2279. doi:10.1109/ICCV.2009.5459452.

[14] D. Glasner, S. Bagon, M. Irani, Super-resolution from a single image, in: IEEE International Conference on Computer Vision (ICCV), 2009, pp. 349–356. doi:10.1109/ICCV.2009.5459271.

[15] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, L. Schumaker (Eds.), Curves and Surfaces, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 711–730.

[16] W. Dong, L. Zhang, G. Shi, X. Li, Nonlocally centralized sparse representation for image restoration, IEEE Transactions on Image Processing 22 (4) (2013) 1620–1630. doi:10.1109/TIP.2012.2235847.

[17] C. Dong, C. C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2) (2016) 295–307. doi:10.1109/TPAMI.2015.2439281.

[18] C. Dong, C. C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 391–407.

[19] Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 370–378. doi:10.1109/ICCV.2015.50.

[20] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2790–2798. doi:10.1109/CVPR.2017.298.

[21] W. Lai, J. Huang, N. Ahuja, M. Yang, Deep laplacian pyramid networks for fast and accurate super-resolution, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5835–5843. doi:10.1109/CVPR.2017.618.

[22] W. Lai, J. Huang, N. Ahuja, M. Yang, Fast and accurate image super-resolution with deep laplacian pyramid networks, CoRR abs/1710.01992. arXiv:1710.01992.
URL http://arxiv.org/abs/1710.01992

[23] M. S. M. Sajjadi, B. Schlkopf, M. Hirsch, Enhancenet: Single image super-resolution through automated texture synthesis, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4501–4510. doi:10.1109/ICCV.2017.481.

[24] Y. Tai, J. Yang, X. Liu, C. Xu, Memnet: A persistent memory network for image restoration, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4549–4557. doi:10.1109/ICCV.2017.486.

[25] K. Zhang, W. Zuo, L. Zhang, Learning a single convolutional super-resolution network for multiple degradations, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3262–3271.

[26] J. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5197–5206. doi:10.1109/CVPR.2015.7299156.

[27] C. Dong, C. C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 184–199.

[28] J. Kim, J. K. Lee, K. M. Lee, Deeply-recursive convolutional network for image super-resolution, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1637–1645. doi:10.1109/CVPR.2016.181.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 2672–2680.

[30] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 694–711.

[31] R. Tsai, T. Huang, Multiframe image restoration and registration, in: Advances in Computer Vision and Image Processing, 1984.

[32] R. R. Schultz, R. L. Stevenson, Extraction of high-resolution frames from video sequences, IEEE Transactions on Image Processing 5 (6) (1996) 996–1011. doi:10.1109/83.503915.

[33] A. J. Patti, M. I. Sezan, A. M. Tekalp, Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time, IEEE Transactions on Image Processing 6 (8) (1997) 1064–1076. doi:10.1109/83.605404.

[34] W. Zhao, H. S. Sawhney, Is super-resolution with optical flow feasible?, in: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), Computer Vision — ECCV 2002, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 599–613.

[35] M. Irani, S. Peleg, Motion analysis for image enhancement: Resolution, occlusion, and transparency, Journal of Visual Communication and Image Representation 4 (4) (1993) 324 – 335. doi:https://doi.org/10.1006/jvci.1993.1030.

[36] C. M. Bishop, A. Blake, B. Marthi, Super-resolution enhancement of video, in: In Proc. Artificial Intelligence and Statistics, 2003.

[37] D. Kong, M. Han, W. Xu, H. Tao, Y. Gong, Video superresolution with scene-specific priors, in: in BMVC, 2006.

[38] Q. Shan, Z. Li, J. Jia, C.-K. Tang, Fast image/video upsampling, ACM Trans. Graph. 27 (5) (2008) 153:1–153:7. doi:10.1145/1409060.1409106.

[39] H. Takeda, P. Milanfar, M. Protter, M. Elad, Super-resolution without explicit subpixel motion estimation, IEEE Transactions on Image Processing 18 (9) (2009) 1958–1975. doi:10.1109/TIP.2009.2023703.

[40] C. Liu, D. Sun, On bayesian adaptive video super resolution, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2) (2014) 346–360. doi:10.1109/TPAMI.2013.127.

[41] Z. Ma, R. Liao, X. Tao, L. Xu, J. Jia, E. Wu, Handling motion blur in multi-frame super-resolution, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5224–5232. doi:10.1109/CVPR.2015.7299159.

[42] Y. Huang, W. Wang, L. Wang, Bidirectional recurrent convolutional networks for multi-frame super-resolution, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems 28, Curran Associates, Inc., 2015, pp. 235–243.

[43] R. Liao, X. Tao, R. Li, Z. Ma, J. Jia, Video super-resolution via deep draft-ensemble learning, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 531–539. doi:10.1109/ICCV.2015.68.

[44] A. Kappeler, S. Yoo, Q. Dai, A. K. Katsaggelos, Video super-resolution with convolutional neural networks, IEEE Transactions on Computational Imaging 2 (2) (2016) 109–122. doi:10.1109/TCI.2016.2532323.

[45] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, W. Shi, Real-time video super-resolution with spatio-temporal networks and motion compensation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2848–2857. doi:10.1109/CVPR.2017.304.

[46] M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, Spatial transformer networks, CoRR abs/1506.02025. arXiv:1506.02025.
URL http://arxiv.org/abs/1506.02025

[47] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, T. Huang, Robust video super-resolution with learned temporal dynamics, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2526–2534. doi:10.1109/ICCV.2017.274.

[48] X. Tao, H. Gao, R. Liao, J. Wang, J. Jia, Detail-revealing deep video super-resolution, in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4482–4490. doi:10.1109/ICCV.2017.479.

[49] X. Mao, C. Shen, Y. Yang, Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections, CoRR abs/1603.09056. arXiv:1603.09056.
URL http://arxiv.org/abs/1603.09056

[50] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, W. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, CoRR abs/1506.04214. arXiv:1506.04214.
URL http://arxiv.org/abs/1506.04214

[51] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks., in: CVPR, Vol. 1, 2017, p. 3.

[52] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, J. Feng, Dual path networks, in: Advances in Neural Information Processing Systems, 2017, pp. 4467–4475.

[53] R. Soltani, H. Jiang, Higher order recurrent neural networks, arXiv preprint arXiv:1605.00064.

[54] W. Wang, X. Li, J. Yang, T. Lu, Mixed link networks, arXiv preprint arXiv:1802.01808.

[55] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European conference on computer vision, Springer, 2016, pp. 630–645.

[56] R. Timofte, E. Agustsson, L. V. Gool, M. Yang, L. Zhang, et al., Ntire 2017 challenge on single image super-resolution: Methods and results, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1110–1121. doi:10.1109/CVPRW.2017.149.

[57] M. Bevilacqua, A. Roumy, C. Guillemot, M. line Alberi Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2012, pp. 135.1–135.10. doi:http://dx.doi.org/10.5244/C.26.135.

[58] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vol. 2, 2001, pp. 416–423 vol.2. doi:10.1109/ICCV.2001.937655.

[59] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, K. Aizawa, Sketch-based manga retrieval using manga109 dataset, Multimedia Tools and Applications 76 (20) (2017) 21811–21838. doi:10.1007/s11042-016-4020-z.
URL https://doi.org/10.1007/s11042-016-4020-z

[60] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980. arXiv:1412.6980.
URL http://arxiv.org/abs/1412.6980

[61] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, 2017.

[62] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612. doi:10.1109/TIP.2003.819861.

[63] M. Haris, G. Shakhnarovich, N. Ukita, Deep back-projection networks for super-resolution, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1664–1673.

[64] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep cnn denoiser prior for image restoration, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2808–2817. doi:10.1109/CVPR.2017.300.

[65] T. Peleg, M. Elad, A statistical prediction model based on sparse representations for single image super-resolution, IEEE Transactions on Image Processing 23 (6) (2014) 2569–2582. doi:10.1109/TIP.2014.2305844.

[66] [link].
URL https://www.pirm2018.org/PIRM-SR.html

[67] C. Ma, C.-Y. Yang, X. Yang, M.-H. Yang, Learning a no-reference quality metric for single-image super-resolution, Computer Vision and Image Understanding 158 (2017) 1–16.

[68] A. Mittal, R. Soundararajan, A. C. Bovik, Making a" completely blind" image quality analyzer., IEEE Signal Process. Lett. 20 (3) (2013) 209–212.

[69] T. Xue, B. Chen, J. Wu, D. Wei, W. T. Freeman, Video enhancement with task-oriented flow, arXiv preprint arXiv:1711.09078.

[70] W. Shi, J. Caballero, F. Huszr, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1874–1883. doi:10.1109/CVPR.2016.207.

[71] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, T. Huang, Robust video super-resolution with learned temporal dynamics, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2526–2534. doi:10.1109/ICCV.2017.274.